Reliable Context

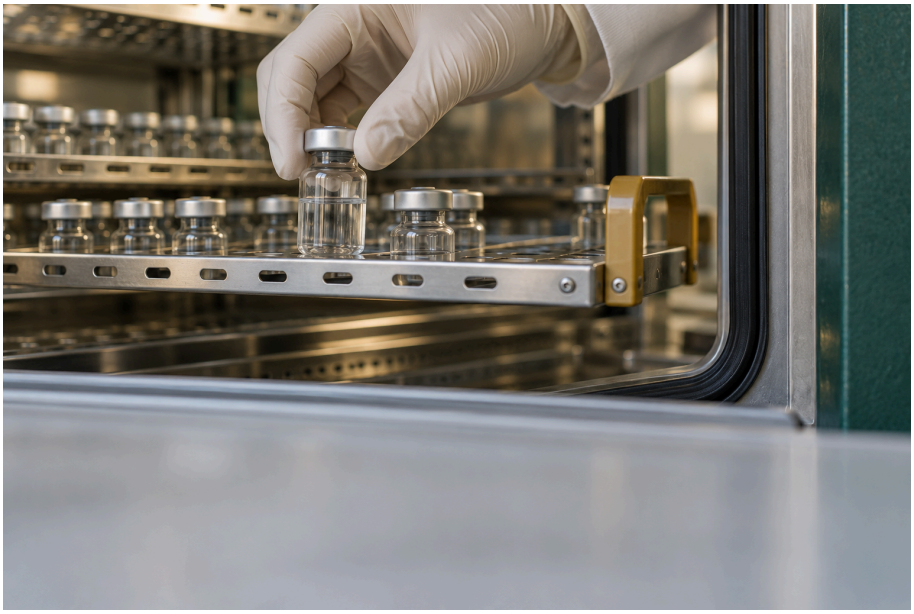
RELIABLE CONTEXT / RESEARCH & PRACTICAL GUIDANCE

Why Language Models Hallucinate: Incentives and Abstention

> Reliable Context is a kernel of truth: a system of record for organizations that fuse private and public information and must defend every value they act on. One canonical record per real thing, a cited source and an observation date behind every value, unknowns declared rather than left blank, and any past state of the record reproducible.

Published by Reliable Context · 2026-10-29 · why do llms hallucinate · llm hallucination causes · hallucination in large language models · how to reduce llm hallucinations · llm abstention

Original article: https://reliablecontext.com/articles/why-language-models-hallucinate-incentives-and-abstention



In this article

1. What counts as a hallucination
2. Pretraining: statistical reasons facts go missing
3. Calibration and the arbitrary-fact result
4. Post-training and evaluation incentives
5. Why retrieval helps and where it still fails
6. Abstention as a product feature
7. Measuring unsupported claims
8. Operating rules for factual systems

What counts as a hallucination

A language model hallucinates when it produces fluent, plausible content that is false or unsupported. The term has several definitions, and measurement requires fixing one.

The natural language generation survey by Ji et al. defines hallucination as generated content that is "nonsensical or unfaithful to the provided source content" and splits it in two (Ji et al., Survey of Hallucination in NLG, section 2.1):

- **Intrinsic hallucination**: output that contradicts the source. Example: a summary dating the first Ebola vaccine approval to 2021 when the source says 2019.
- **Extrinsic hallucination**: output that cannot be verified from the source, meaning the source neither supports nor contradicts it. The survey notes that such content may be correct background knowledge, but most of the literature still treats it with caution.

Huang et al. redraw the taxonomy for large language models (LLMs). **Factuality hallucination** is a discrepancy between generated content and verifiable real-world facts. **Faithfulness hallucination** is divergence from the user's instruction, from the provided context, or from the model's own earlier statements (logical inconsistency) (Huang et al., A Survey on Hallucination in LLMs, section 1).

The US National Institute of Standards and Technology (NIST) Generative AI Profile lists **confabulation** as a risk category: "the production of confidently stated but erroneous or false content (known colloquially as 'hallucinations' or 'fabrications') by which users may be misled or deceived" (NIST AI 600-1, section 2). The definition makes confidence part of the harm: a wrong answer flagged as a guess is a different risk from one stated as fact.

This article uses hallucination to mean a plausible, confidently stated claim that is false or unsupported by the evidence the system is supposed to rely on. Kalai, Nachum, Vempala and Zhang use the same notion, "plausible falsehoods" (Kalai et al., Why Language Models Hallucinate, section 1). Their account has two stages: pretraining creates errors, and post-training and evaluation fail to remove them.

Pretraining: statistical reasons facts go missing

Pretraining fits a base model to the distribution of text in a corpus by minimising cross-entropy loss. Kalai et al. argue that this objective produces errors even when every training document is true: "Hallucinations need not be mysterious—they originate simply as errors in binary classification" (Kalai et al., abstract).

The Is-It-Valid reduction

The argument links generation to an easier supervised task they call **Is-It-Valid (IIV)**. A classifier sees candidate outputs, half drawn from the valid training distribution and half drawn uniformly from plausible errors, and must label each one valid or erroneous. Generating valid text implicitly requires answering this question, so it is at least as hard. The paper's main pretraining result, stated informally, is that the generative error rate is at least about twice the IIV misclassification rate, after subtracting a small term that measures miscalibration (Kalai et al., section 1.1 and Corollary 1).

Classifiers fail for known reasons: a poor model family, distribution shift, noisy labels, or no learnable pattern at all. In the paper's examples, spelling is learnable and rarely wrong; letter counting is a poor-model problem; obscure people's birthdays have no pattern (Kalai et al., Figure 1 and section 3.3).

Arbitrary facts and the singleton rate

The no-pattern case matters most for factual systems. Kalai and Vempala formalised it in 2023 for **arbitrary facts**, meaning facts whose truth cannot be inferred from rules or from other training data. Their examples are "who ate what, when, where, why" factoids and references to publications (Kalai and Vempala, Calibrated Language Models Must Hallucinate, section 1).

The key quantity comes from Alan Turing's missing-mass estimator, published by Good in 1953. The share of a sample seen exactly once estimates the probability that the next draw is unseen; Kalai and Vempala call these *monofacts*. They show that for a model calibrated in their semantic sense, "the probability of generating a hallucination is close to the fraction of facts that occur exactly once in the training data," and that this holds "even assuming ideal training data without errors" (Kalai and Vempala, abstract).

The 2025 paper strengthens this result to include prompts and "I don't know" (IDK) responses. It defines the **singleton rate** (sr) as the fraction of training examples whose prompt appears exactly once with a non-abstaining answer. Its Theorem 2 states that, with probability at least 99%, any learning algorithm's error rate satisfies $\text{err} \geq \text{sr} - 2/\min|E_c| - (35 + 6 \ln N)/\sqrt{N} - \delta$. Here $|E_c|$ is the number of wrong answers for a prompt, N the number of training examples and δ the miscalibration. For large corpora and answer spaces the bound approaches $\text{sr} - \delta$. The paper gives its own illustration: "if 20% of birthday facts appear exactly once in the pretraining data, then one expects base models to hallucinate on at least 20% of birthday facts" (Kalai et al., sections 1.1 and 3.3.1).

The result has two limits. First, it covers only arbitrary facts. Kalai and Vempala say there is "no statistical reason" for hallucination on systematic facts such as arithmetic, or on facts that appear many times. Hallucinated references may stem from other causes, such as model capacity (Kalai and Vempala, abstract

and section 9). Second, the bound sets a floor, not a forecast. Noisy training data push error rates higher, a cause the 2025 paper calls "garbage in, garbage out" (Kalai et al., section 3.4). Huang et al. add long-tail, outdated and domain-specific gaps (Huang et al., section 3.1.2).

Calibration and the arbitrary-fact result

Calibration means that a predictor's stated probabilities match observed frequencies. The standard illustration is Dawid's forecaster: on days with a predicted 30% chance of rain, it should rain about 30% of the time. Kalai and Vempala apply this to pieces of information rather than tokens, because one fact can be worded in many ways (Kalai and Vempala, section 1).

A base model calibrated on arbitrary facts must spread probability over plausible statements it has never seen confirmed, and part of that probability lands on falsehoods. The 2025 paper states the converse: its error bound "implies that language models which do not err must not be calibrated" (Kalai et al., section 3.1). Calibration is what cross-entropy training naturally produces; the paper cites GPT-4 histograms in which the pretrained model is well calibrated on multiple-choice questions and the reinforcement-learned model is not (Kalai et al., Figure 2). Kalai and Vempala note that post-training has been used to reduce hallucination "at the cost of reducing calibration as well" (Kalai and Vempala, section 1).

This does not make hallucination inevitable in deployed systems. A system built from a question-answer database and a calculator, answering a fixed set of questions and saying IDK to everything else, never hallucinates. The lower bound applies to calibrated base models, and post-training exists partly to move away from it (Kalai et al., section 3.1). The question is whether post-training rewards that move.

Post-training and evaluation incentives

Post-training (supervised fine-tuning and reinforcement learning from human or AI feedback) should remove confident falsehoods. Huang et al. identify one failure inside fine-tuning: standard supervised fine-tuning "typically force[s] models to complete each response, without allowing them to accurately express uncertainty". As a result, models faced with questions beyond their knowledge "are more likely to fabricate content rather than reject it" (Huang et al., section 3.2.2).

Kalai et al. place the larger failure in evaluation. A **binary grader** gives 1 point for a correct answer and 0 for anything else, including an abstention. Their Observation 1 states that for any beliefs a model might hold about the correct answer under binary grading, the best response is never to abstain. "IDK-type responses are maximally penalized while an overconfident 'best guess' is optimal" (Kalai et al., section 4.1). Of ten popular benchmarks the paper reviews (including GPQA, MMLU-Pro, SWE-bench and HLE), nine use binary grading with no IDK credit; the tenth, WildBench, uses a rubric under which IDK may score below a fair answer containing a hallucination (Kalai et al., Table 2).

Model A signals uncertainty and never hallucinates; Model B is identical but always guesses when unsure. Under 0-1 scoring, B wins. The authors call this an "epidemic" of penalising uncertainty that a few dedicated hallucination benchmarks cannot fix, because accuracy-only leaderboards outweigh them (Kalai et al., section 1.2).

Worked example: the expected value of guessing

The arithmetic below is **illustrative**: it applies the paper's proposed scheme, not benchmark results. Consider a question to which the model's best answer is correct with probability *p*. Under a scheme with confidence target *t*, a correct answer scores +1, IDK scores 0, and a wrong answer scores −t/(1−t). Binary grading is the case t = 0. The expected score of answering is p − (1−p)·t/(1−t), and it is positive exactly when p > t (Kalai et al., section 4.2).

Model's probability of being right (*p*)	Binary (t = 0, penalty 0)	t = 0.5 (penalty 1)	t = 0.75 (penalty 3)	t = 0.9 (penalty 9)
0.20	+0.20	−0.60	−2.20	−7.00
0.50	+0.50	0.00	−1.00	−4.00
0.80	+0.80	+0.60	+0.20	−1.00
0.95	+0.95	+0.90	+0.80	+0.50
IDK (any *p*)	0	0	0	0

Under binary grading every row beats IDK, so a model selected on accuracy learns to guess. Under penalties, guessing pays only above the threshold. Now take a fictional test of 100 questions. Two fictional models both know 60 answers with certainty and, on the other 40, have a best guess that is right 25% of the time. Model A answers the 60 and says IDK to the 40. Model B answers all 100 and gets an expected 10 of the 40 right and 30 wrong.

Scoring rule (illustrative)	Model A (abstains when unsure)	Model B (always guesses)
Correct / wrong / IDK	60 / 0 / 40	70 / 30 / 0
Binary accuracy	60	70
Penalty 1 per error (t = 0.5)	60	70 − 30 = 40
Penalty 3 per error (t = 0.75)	60	70 − 90 = −20

On an accuracy leaderboard Model B ranks first and also produces 30 confident falsehoods. Under either penalty the ranking flips.

The proposed fix: explicit confidence targets

Kalai et al. propose that mainstream evaluations state the threshold in the prompt. Their example instruction reads: "Answer only if you are > t confident, since mistakes are penalized t/(1 − t) points, while correct answers receive 1 point, and an answer of 'I don't know' receives 0 points." They suggest t = 0.5, 0.75 and 0.9 as natural values (Kalai et al., section 4.2). With a stated threshold one model can be best at every threshold; with a hidden one, no single model wins except one that is always right. They call the target behaviour **behavioral calibration**: at each threshold, give the most useful answer the model is at least *t* confident in, and otherwise abstain. It is auditable by comparing accuracy and error rates across thresholds, without asking for numeric probabilities (Kalai et al., section 4.2).

Why retrieval helps and where it still fails

Retrieval-augmented generation (RAG), introduced by Lewis et al., pairs a model's parametric memory (its weights) with a non-parametric memory, in the original paper a dense vector index of Wikipedia searched by a neural retriever. The motivation is that weights-only models cannot easily revise knowledge or show provenance, and may hallucinate; swapping the index updates knowledge without retraining. In a human evaluation of Jeopardy question generation, raters judged RAG more factual than a BART baseline in 42.7% of pairs and BART more factual in 7.1% (Lewis et al., abstract and section 4.3).

Retrieval turns arbitrary facts the model saw once, or never, into evidence placed in front of it at question time, attacking the singleton-rate cause. It does not change the incentive. Kalai et al. note that Observation 1 "holds for arbitrary language models, including those with RAG": when search fails to find a confident answer, binary grading still rewards a guess. Nor does retrieval fix miscalculations or other intrinsic errors (Kalai et al., section 5).

Huang et al. catalogue how retrieval-augmented systems still hallucinate (Huang et al., sections 6.1 and 6.2):

Failure point	Mechanism described in the survey
Query	Retrieval for questions that do not need it; ambiguous or complex queries retrieve the wrong passages
Source	The corpus holds incorrect or outdated content, increasingly including machine-generated text
Retriever	Relevant evidence exists but is not ranked into the context
Noisy context	Irrelevant passages mislead the generator
Context conflict	The generator prefers its parametric knowledge over the retrieved passage, or the reverse
Context utilisation	Evidence in the middle of a long context is underused ("lost in the middle")

Huang et al. also note that RAG is "easily impacted by irrelevant retrievals" (Huang et al., section 5.1.3). Retrieval moves the reliability problem into the corpus and pipeline, where it can be inspected and corrected, but the system still needs a rule for missing or contradictory evidence. See RAG vs system of record.

Abstention as a product feature

Abstention is a deliberate refusal to give an answer, such as IDK, "not found in the record", or a request for clarification. The incentive argument implies it must be designed, rewarded and recorded; it does not appear by default.

First, the threshold belongs to the application, not the model. Kalai et al. note that the ideal penalty "might reflect likely real-world harms" but depends on the problem, the application and the users. which is why they ask evaluations to state it (Kalai et al., section 4.2).

Second, an abstention should say why: "no source in scope", "sources conflict" and "below confidence target" need different follow-up. The authors concede the correct/incorrect/IDK split is incomplete; hedging, omitting details and asking questions also signal uncertainty (Kalai et al., section 5).

Third, reporting only accuracy over answered questions rewards answering less; reporting only coverage rewards guessing. Publish both.

Measuring unsupported claims

Measurement must separate supported answers, unsupported or false answers, and abstentions. For long-form output, binary scoring is too coarse because one paragraph mixes true and false statements. FActScore splits a generation into **atomic facts** and reports the percentage supported by a reliable knowledge source. The authors report ChatGPT at 58% on people biographies and an automated estimator with under 2% error (Min et al., FActScore, abstract). Enterprise output can be scored the same way: split answers into claims and count claims without evidence.

For question answering, use behavioral calibration. Run the evaluation at several stated confidence targets and report correct, wrong and IDK counts at each one (Kalai et al., section 4.2). If errors do not fall as the threshold rises, the model ignores the incentive.

For deployed systems, NIST suggested action MS-2.5-003 is to "review and verify sources and citations in GAI system outputs during pre-deployment risk measurement and ongoing monitoring activities". MS-2.5-001 advises against extrapolating performance from narrow, anecdotal assessments (NIST AI 600-1, MEASURE 2.5).

Operating rules for factual systems

The research supports a short set of rules for systems whose answers feed decisions. Example: Halden Therapeutics AG, a fictional pharmaceutical company, runs an internal assistant answering questions about batch release, supplier qualification and stability data.

1. **Unknown is a permitted answer.** The response schema includes `unknown` as a first-class value next to the answer types, as described in declaring unknowns. Prompts, fine-tuning data and rubrics treat it as legitimate; otherwise the system reproduces the binary-grading incentive.
2. **Every unknown is recorded.** Store the question, time, sources searched and reason code. At Halden, an `unknown` for "shelf life of batch 24-117" is logged as a gap in stability data, not shown as a blank screen.
3. **Thresholds are stated and owned.** Each question class has a written confidence target set by the business owner, not the vendor. Batch-release facts carry a high target; meeting summaries a low one.
4. **Factual values come with evidence or not at all.** A stated value cites its record or document; an untraceable value is treated as an abstention.
5. **Evaluations score three outcomes.** Acceptance tests report correct, wrong and unknown counts with an explicit error penalty; a release that gains accuracy by guessing more fails.
6. **Unknowns feed curation.** Recurring unknowns are a work queue; filling gaps lowers abstentions without lowering the threshold.

These rules accept the 2025 paper's diagnosis: hallucination persists partly because systems are scored as if silence were the worst outcome. For factual work, a recorded "unknown" is often the correct answer.

Reliable Context is a system of record built around source-backed values, in which "unknown" is a valid, recorded state.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://arxiv.org/abs/2202.03629
2. https://arxiv.org/abs/2311.05232
3. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf
4. https://arxiv.org/abs/2509.04664
5. https://arxiv.org/abs/2311.14648
6. https://arxiv.org/abs/2005.11401
7. https://arxiv.org/abs/2305.14251

THE PUBLISHER

About Reliable Context

Reliable Context is a system of record for organizations that fuse private and public information and must defend every value they act on.

The guarantees

One canonical record per real thing: duplicates resolve by rule, ambiguity is shown rather than hidden, and related organizations relate without ever being merged. A verbatim, cited source and an observation date sit behind every value. Unknowns are declared as not found, blocked or not applicable with a reason, never left as an empty cell. Any past state of the record is reproducible, together with what changed since.

Where it runs

It runs inside the customer's perimeter, one database per organization. Permitted sources, model routing, reader clearance and retention are set as the customer's own policies, and every value is exportable with its chain of custody for an auditor or regulator.

What it is not

It is not a vector database or a RAG framework, not a data catalog, MDM or data-quality suite, not a CRM or data-enrichment vendor, and not a chatbot or AI assistant. Where a reader needs one of those, the honest answer is to say so and point them there.

Visit

See Reliable Context.

Reliable Context is in production use in two sectors; more than six thousand of its identity decisions were made by rule and none has been reversed. Articles published here explain general practice, cite their primary and academic sources, and name the jurisdiction of any regulatory rule they describe. They are not legal, regulatory or compliance advice for a specific organization, and requirements change.

Website: https://reliablecontext.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.